Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Engineering 29 (2012) 1867 – 1871

Procedia Engineering

www.elsevier.com/locate/procedia

2012 International Workshop on Information and Electronics Engineering (IWIEE)

The Hardware-in-the-loop Simulation on the Control System of a Small Launch Vehicle

Huang Wenbo[a]*, Zhang Qiang[b]

[a]College of Aerospace and Material Engineering, National University of Defense Technology, Changsha 410073,China
[b]Army Aviation Institute, Beijing 101123, China

Abstract

The hardware-in-the-loop simulation plays an important role in aerocraft design. In hardware-in-the-loop simulation on launch vehicle, some hardware components are connected to the simulation loop and then work driven by data from the running computer which yield different operand quantities in different coordinate frames. According to the analysis on characteristic of the ballistics in the inertial launch reference frame, a hardware-in-the-loop simulation system for a small launch vehicle containing IMU, rocket-borne computer and rudder is established. Simulation results indicate the control scheme is accurate, the algorithm of navigation, guidance and attitude control are valid, and the calculation reduction quantity benefited from using the data in the inertial launch reference frame rather than in launch reference is worked out.

Keywords: small launch vehicle; hardware-in-the-loop simulation; rocket-borne computer; control algorithm

1. Introduction

Hardware-in-the-loop simulation is a well established technique used to validate the correctness of the navigation algorithm[1] and guidance algorithm [2], and to guide the design of the controller[3]. In this way, the work process of the hardware in the control system can recur in various flight conditions, so it plays an important roles in the controller design[4,5]. This paper firstly analyzes the operand quantities of the rocket-borne computer when the navigation reference frame is the launch inertial frame, and then makes clear the data transfer process of the simulation system, lastly, the hardware-in-the-loop simulation

* Corresponding author. Tel.: +86-0731-84576482; fax:.+86-0731-8412301
E-mail address:hwbnudt@gmail.com

doi:10.1016/j.proeng.2012.01.228

system for the control system of a small launch vehicle is established. By simulating the deviation case after the first stage's separate, the ability of the control system is obtained.

2. Operand quantities in rocket-borne computer

2.1. Flight program

The rockets arise within the atmosphere usually in the perturbation guidance way that based the flight program. The data of flight program is stored in the rocket-borne computer, which is designed as follows[6]: $\varphi_T = \varphi_{pr}(t)$, $\psi_T = \gamma_T = 0$. φ_A , ψ_A , γ_A are the euler angles, respectively the pitch angle, yaw angle and roll angle in the launch inertial frame. The relationship between euler angles in the launch inertial frame and in the launch frame is in this formula:

$$\begin{cases} \varphi(t) = \varphi_{pr} - \omega_{ez} t \\ \psi(t) = -(\omega_{ey} \cos\varphi - \omega_{ex} \sin\varphi) t \\ \gamma(t) = -(\omega_{ey} \sin\varphi + \omega_{ex} \cos\varphi) t \end{cases} \tag{1}$$

Where φ , ψ , γ are the euler angles relative to the launch frame, ω_{ex} , ω_{ey} , ω_{ez} are the rotation ration of the earth at the launch point in the launch inertial frame.

It is seen the flight program in the launch inertial frame is simpler than that in the launch frame. The flight program in yaw and roll channels is zero all the time in the launch inertial frame, but flight program vary with time in the launch frame. So it undoubtedly increases the computation of the rocket-borne computer.

2.2. Navigation solution

The dynamic equation in the launch frame is as follows:

$$\frac{\delta^2 \mathbf{r}}{\delta t^2} = \dot{\mathbf{W}} + \mathbf{g} - \boldsymbol{\omega}_e \times (\boldsymbol{\omega}_e \times \mathbf{r}) - 2\boldsymbol{\omega}_e \times \frac{\delta \mathbf{r}}{\delta t} \tag{2}$$

While it is simplified as follows in the launch inertial frame[6]:

$$\frac{d^2 \mathbf{r}}{dt^2} = \dot{\mathbf{W}} + \mathbf{g} \tag{3}$$

Where $\dot{\mathbf{W}}$ is the visual acceleration measured by the accelerometers. Comparing the former two formulas, it is obviously that navigation solution operation in the launch frame is greater than the other one, because the last two items in formula (2) need to be corrected.

2.3. Orbit parameter calculation

It is required to calculate the real-time orbit parameters when the launch vehicle is close to the point in the designed orbit, to compare with the ideal orbit parameters. The orbital parameters are described as the six elements, which is easier to be calculated from data of the launch inertial frame than the launch frame.

Taking $M_i[\theta]$ to represent the elementary transformation matrix, the transformation matrix from inertial coordinate frame to launch frame is:

$$\begin{aligned} I_G = M_3[\pi/2 - (\lambda_0 + S_0 + \omega_e t)] \\ M_1[-\varphi_0] M_2[\pi/2 + \alpha_0] \end{aligned} \tag{4}$$

where λ_0 is the longitude of the launch point, φ_0 is the geocentric latitude of the launch point, S_0 is the Greenwich mean sidereal time (GMST), ω_e is the earth rotation rate, t is the time from launch.

The transformation matrix from inertial coordinate frame to launch inertial frame is:

$$I_A = M_3[\pi/2-(\lambda_0+S_0)] \\ M_1[-\varphi_0]M_2[\pi/2+\alpha_0] \quad (5)$$

The transformation matrix I_A is a constant matrix which is counted once, but I_G is a matrix varied with time. Clearly, the former may reduce the amount of computation of the rocket-borne computer.

3. Hardware-in-the-loop simulation of a small launch vehicle

3.1. Structure of the simulation system

The hardware-in-the-loop simulation system consists of the DSPACE simulator, IMU unit, three-axis turntable, DSP-based on-board computer, servo, cables, power supply and other auxiliary equipment composition. The sketch map of the simulation system is shown in Fig 1.

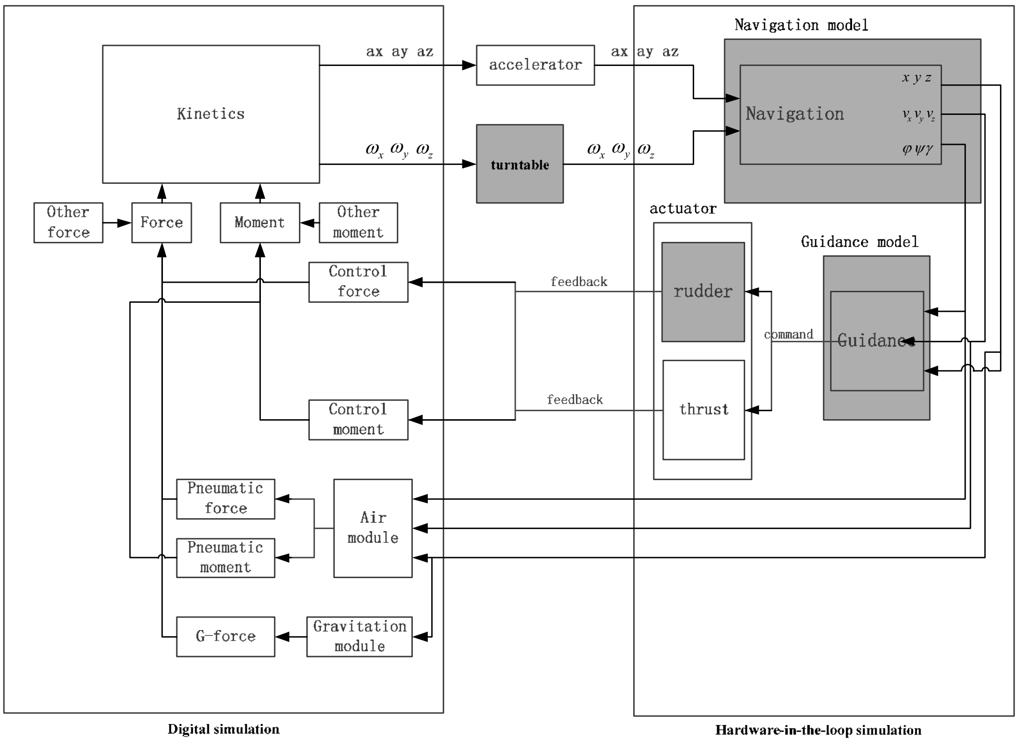
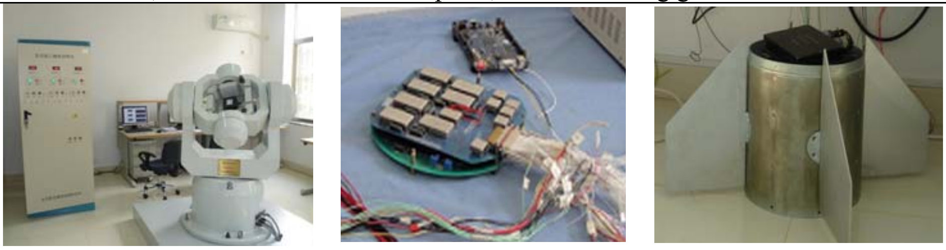


Fig 1 Sketch map of the simulation system

3.2. Hardware in the simulation system

In Fig 1, the shaded modules are the physical devices in the simulation, and the transparent ones are the digital simulation modules. The digital simulation models are successfully compiled on the host computer and then running in the simulator -DS1005 processor after downloaded. The navigation, guidance and control algorithms are developed using standard C, downloaded to the DSP chip in the onboard computer rocket. Parts of physical devices in the simulation circuit are shown in Fig 2, where a is the three-axis turntable, b is the rocket borne computer, c is the steering gear.



a b c

Fig 2 Physical devices in the simulation loop

3.3. *Data flow in the simulation*

The data flow of the simulation is as follows:

① The kinetic module generates the angular ration and the acceleration relative to the body from the current states; ② The data passed to the three-axis turntable, then it drive the installed IMU unit rotate and output the rotation information; ③ The navigation module calculates the current state features using the information. The state features are passed to the guidance module; ④ The guidance module calculate the guidance instruction according to the guidance law, then the guidance instruction is passed to the actuator; ⑤ The actuators calculates the force and torque, and produces feedback to the rocket; ⑥ The ballistics calculation module integrates the kinematics and dynamics equations, gets the state parameters of the next step, completing the closed simulation loop.

4. Simulation results

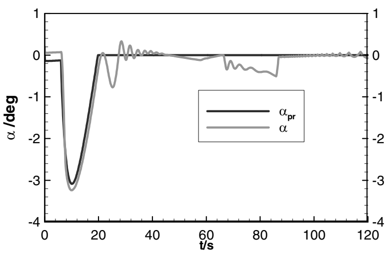


Fig 3 The angle of attack of desired/actual flight trajectory

To simulate the flight phase of a small two-stage launch vehicle in the established simulation system. The first separation time is 65s, and then the coast phase starts, the second stage ignites at 92s. The angle of attack of the desired trajectory and the bias trajectory under separation deviation of first stage is shown in Fig 3. As the figure shows, at the end of the turn phase of the negative angle of attack (20s), angle of attack began to oscillate, after a short time of control, the amplitude tends to be zero. At the time of 65s, the body posture error occurs because of the separation deviation, but the angle of attack has been controlled nearby the designed valve before the coast segment end (92s), and the posture has been stable when the second stage starts to work. It is seen that the control system accomplish the mission of control the posture.

The actual orbit parameters and orbit parameters error at injection are shown in Table 1, the orbit deviations meet the needs of the overall program.

Table 1 Orbit parameters and error at injection

Orbital elements	a /m	e	i /deg	Ω /deg	ω /deg	τ /s
Ideal parameters	6690032	0.0059	49.9780	225.9988	11.8515	-460.429
Actual parameter	6692685	0.0072	50.0250	226.0052	12.1071	-453.064
error	2653	0.0013	0.0470	0.0063	0.2556	7.365

In the table, a , e , i , Ω , ω , τ is respectively semi-major axis, eccentricity, inclination, ascension of ascending node, angle from the perigee and perigee time of the orbit.The reduction of operations of the rocket-borne computer in the launch inertial frame is shown in Table 2, n is the total number of the calculation in the flight, decided both by the flight time and the step of calculation.

Table 2 Amount of computation reduction of the rocket-borne computer

Calculation type	plus/ subtraction	multiplication / division	sine
Flight program	5n	7n	4n
Navigation	15n	18n	-
Orbit parameter	n	n	3n

5. Conclusion

This paper has analyzed the characteristics of the trajectory in the launch inertial frame and established the hardware-in-the-loop simulation system. Simulation results show that this project of simulation system is feasible, and the navigation, guidance and control algorithms in the onboard computer are correct. By quantitatively comparing the calculate operations in the launch inertial frame and in the launch frame during the flight of a small launch vehicle, it is concluded that it not only reduce the data storage in the rocket-borne computer, but also reduce the operand quantities of the navigation solution and orbit point parameter calculation algorithms based the trajectory in the launch inertial frame.

References

[1] Quan W, Fang J. Hardware in-the-loop simulation of celestial navigation system. AIAA Modeling and Simulation Technologies Conference and Exhibit. San Francisco, California, 15-18 August, 2005.

[2] Marko B, Monte M. Hardware-in-the-loop simulation of aerodynamic objects. AIAA Modeling and Simulation Technologies Conference and Exhibit. Hilton Head, South Carolina, 20-23 August, 2007.

[3] Tam H, Liu HHT, Rabbath CA. Hardware-In-The-Loop Simulation for an UAV Example. AIAA Modeling and Simulation Technologies Conference and Exhibit. San Francisco, California, 15-18 August, 2005.

[4] Michael S, Michael JP. Hardware in the Loop Simulation of Embedded Automotive Control Systems. Proceedings of the 8th International IEEE Conference on Intelligent Transportation Systems. Vienna, Austria, September 13-16, 2005.

[5] Bacic M. On hardware-in-the-loop simulation. Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference 2005. Seville, Spain, December 12-15, 2005.

[6] Yang Xi-Xiang,Zhang Wei-Hua,Xiao Fei,et al. Launching capability analysis of small solid launch vehicle[J]. Journal of Solid Rocket Technology. 2009, 32(4): 355-359.